

IN THE CLAIMS:

1. (Currently Amended) A method for providing at least one self-tuning object to a user program, the method comprising:

receiving said user program;

simulating execution of said user program;

detecting, during said simulation of said execution of said user program, occurrences ~~a plurality of expressions including~~ using said at least one self-tuning object in said user program,

generating, for each occurrence, in response to said detecting ~~said plurality of expressions including said at least one self tuning object in said user program~~, an entry in a trace file including data representing said expressions and indicating the sequence in which said expressions occur reflecting an execution flow of said expressions in said user program during said simulating and enabling generation of source code corresponding to said expressions;

dividing said trace file into a plurality of trace file blocks;

converting said trace file blocks into source code expression blocks;

generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating including, for each source code expression block,

parameterizing said source code expression block to include at least one ~~of a loop blocking parameter and a loop unrolling~~ optimization parameter, the at least one optimization parameter being taken from parameters of self-tuning objects corresponding to entries in a trace file block from which said source code expression block was generated,

iteratively:

selecting at least one value for said ~~respective~~ at least one optimization parameter,

compiling said parameterized source code expression block in accordance with said selected at least one value for said ~~respective~~ at least one optimization parameter, and

measuring an execution time of object code resulting from that compiling, and,

on the basis of iteratively selecting, compiling and measuring, identifying the at least one value for said ~~respective~~ at least one optimization parameter that is associated with ~~the~~ a minimal ~~said~~ execution time for said compiled expression block; and,

linking said plurality of minimal timing, compiled expression blocks into said user program.

2. (Currently Amended) The method of claim 1, wherein said detecting said ~~plurality~~ occurrences of expressions ~~including using~~ said at least one self-tuning object in said user program is performed by program code associated with at least one overloaded operator associated with said at least one self-tuning object.

3. (Currently Amended) The method of claim 1, wherein said generating a trace file ~~indicating the sequence~~ reflecting an execution flow of said expressions ~~including using~~ said at least one self-tuning object in said user program is performed by program code associated with at least one overloaded operator associated with said at least one self-tuning object.

4. (Original) The method of claim 1, wherein said dividing said trace file into said plurality of trace file blocks is performed such that a total amount of computational dependencies and synchronization requirements within said user program, including computational dependencies and synchronization requirements between trace file blocks, are minimized.

5. (Original) The method of claim 1, wherein said dividing said trace file into said plurality of trace file blocks is performed responsive to user provided delimiters included within said user program.

6-7. (Cancelled)

8. (Original) The method of claim 1, wherein said linking of said minimal timing, compiled expression blocks to said user program is responsive to execution of said user program.

9. (Currently Amended) The method of claim 8, wherein said linking of said minimal timing, compiled expression blocks further comprises detecting, during said execution of said user program, ~~said plurality~~ corresponding occurrences of expressions ~~including using~~ said at least one self-tuning object in said user program.

10. (Original) The method of claim 9, wherein said linking of said minimal timing, compiled expression blocks further comprises scheduling said minimal timing, compiled expression blocks for execution on at least one processor of a target parallel processing computer.

11. (Currently Amended) A computer program product including a computer readable medium, said computer readable medium having at least one computer program stored thereon, said at least one computer program comprising:

program code for receiving said user program;

program code for simulating execution of said user program;

program code for detecting, during said simulation of said execution of said user program, occurrences ~~a plurality of~~ expressions ~~including using~~ said at least one self-tuning object in said user program;

program code for generating, for each occurrence, in response to said detecting ~~said plurality of expressions including said at least one self-tuning object in said user program, an entry in~~ a trace file including data representing said expressions and ~~indicating the sequence in which said expressions occur~~ reflecting an execution flow of said expressions in said user program during said simulating and enabling generation of source code corresponding to said expressions;

program code for dividing said trace file into a plurality of trace file blocks;

program code for converting said trace file blocks into source code expression blocks;

program code for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating including, for each source code expression block,

parameterizing said source code expression block to include at least one of a ~~loop blocking parameter and a loop unrolling~~ optimization parameter, ~~the at least one optimization parameter being taken from parameters of self-tuning objects corresponding to entries in a trace file block from which said source code expression block was generated,~~

iteratively:

selecting at least one value for said ~~respective~~ at least one optimization parameter,

compiling said parameterized source code expression block in accordance with said selected at least one value for said ~~respective~~ at least one optimization parameter, and

measuring an execution time of object code resulting from that compiling, and,

on the basis of iteratively selecting, compiling and measuring, identifying the at least one value for said ~~respective~~ at least one optimization parameter that is associated with ~~the~~ a minimal ~~said~~ execution time for said compiled expression block; and,

program code for linking said plurality of minimal timing, compiled expression blocks into said user program.

12. (Currently Amended) The computer program product of claim 11, wherein said program code for detecting said ~~plurality~~ occurrences of expressions ~~including~~ using said self-tuning object in said user program comprises program code associated with at least one overloaded operator associated with said self-tuning object.

13. (Currently Amended) The computer program product of claim 11, wherein said program code for generating a trace file ~~indicating the sequence~~ reflecting an execution flow of said expressions ~~including~~ using said at least one self-tuning object in said user program

comprises program code associated with at least one overloaded operator associated with said at least one self-tuning object.

14. (Original) The computer program product of claim 11, wherein said program code for dividing said trace file into said plurality of trace file blocks is operative to divide said trace file into said plurality of trace file blocks such that a total amount of computational dependencies and synchronization requirements within said user program, including computational dependencies and synchronization requirements between trace file blocks, are minimized.

15. (Original) The computer program product of claim 11, wherein said program code for dividing said trace file into said plurality of trace file blocks is operative to divide said trace file into said plurality of trace file blocks responsive to user provided delimiters included within said user program.

16-17. (Cancelled)

18. (Original) The computer program product of claim 11, wherein said program code for linking of said minimal timing, compiled expression blocks to said user program is triggered by execution of said user program.

19. (Currently Amended) The computer program product of claim 18, wherein said linking of said minimal timing, compiled expression blocks further comprises program code for detecting, during said execution of said user program, said plurality corresponding occurrences of expressions including using said at least one self-tuning object in said user program.

20. (Original) The computer program product of claim 19, wherein said program code for linking of said minimal timing, compiled expression blocks further comprises program code for scheduling said minimal timing, compiled expression blocks for execution on at least one processor of a target parallel processing computer.

21. (Original) The computer program product of claim 11, wherein said computer program comprises a compiler.

22. (Currently Amended) A computer data signal embodied in a carrier wave, said computer data signal including at least one computer program, said at least one computer program comprising:

program code for receiving said user program;

program code for simulating execution of said user program;

program code for detecting, during said simulation of said execution of said user program, occurrences ~~a plurality of expressions including~~ using said at least one self-tuning object in said user program;

program code for generating, for each occurrence, in response to said detecting ~~said plurality of expressions including said at least one self-tuning object in said user program, an entry in~~ a trace file including data representing said expressions and ~~indicating the sequence in which said expressions occur~~ reflecting an execution flow of said expressions in said user program during said simulating and enabling generation of source code corresponding to said expressions;

program code for dividing said trace file into a plurality of trace file blocks;

program code for converting said trace file blocks into source code expression blocks;

program code for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating including, for each source code expression block,

parameterizing said source code expression block to include at least one ~~of a loop blocking parameter and a loop unrolling~~ optimization parameter, ~~the at least one optimization parameter being taken from parameters of self-tuning objects corresponding to entries in a trace file block from which said source code expression block was generated,~~

iteratively:

selecting at least one value for said ~~respective~~ at least one optimization parameter,
compiling said parameterized source code expression block in accordance with said selected at least one value for said ~~respective~~ at least one optimization parameter, and
measuring an execution time of object code resulting from that compiling, and,
on the basis of iteratively selecting, compiling and measuring, identifying the at least one value for said ~~respective~~ at least one optimization parameter that is associated with ~~the~~ a minimal ~~said~~ execution time for said compiled expression block; and,
program code for linking said plurality of minimal timing, compiled expression blocks into said user program.

23. (Currently Amended) A system for providing at least one self-tuning object to a user program, the system comprising:

at least one processor;
at least one memory communicably coupled to said at least one processor;
a computer program for execution on said processor, said computer program stored
in said memory, said computer program comprising:
program code for receiving said user program;
program code for simulating execution of said user program;
program code for detecting, during said simulation of said execution of said user program, occurrences a plurality of expressions including using said at least one self-tuning object in said user program;
program code for generating, for each occurrence, in response to said detecting ~~said plurality of expressions including said at least one self-tuning object in said user program, an entry in a trace file including data representing said expressions and indicating the sequence in which said expressions occur reflecting an execution flow of said expressions~~ in said user

program during said simulating and enabling generation of source code corresponding to said expressions;

program code for dividing said trace file into a plurality of trace file blocks;

program code for converting said trace file blocks into source code expression blocks;

program code for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating including, for each source code expression block,

parameterizing said source code expression block to include at least one of a ~~loop blocking parameter and a loop unrolling~~ optimization parameter, ~~the at least one optimization parameter being taken from parameters of self-tuning objects corresponding to entries in a trace file block from which said source code expression block was generated,~~

iteratively:

selecting at least one value for said ~~respective~~ at least one optimization parameter,

compiling said parameterized source code expression block in accordance with said selected at least one value for said ~~respective~~ at least one optimization parameter, and

measuring an execution time of object code resulting from that compiling, and,

on the basis of iteratively selecting, compiling and measuring, identifying the at least one value for said ~~respective~~ at least one optimization parameter that is associated with ~~the~~ a minimal ~~said~~ execution time for said compiled expression block; and,

program code for linking said plurality of minimal timing, compiled expression blocks into said user program.

24. (Currently Amended) A system for providing at least one self-tuning object to a user program, comprising:

means for receiving said user program;

means for simulating execution of said user program;

means for detecting, during said simulating of said execution of said user program, a ~~plurality occurrences~~ of expressions ~~including~~ using said at least one self-tuning object in said user program;

means for generating, for each occurrence, in response to said detecting ~~said plurality of expressions including said at least one self-tuning object in said user program~~, an entry in a trace file including data representing said plurality of expressions and indicating the sequence in which said expressions occur reflecting an execution flow of said expressions in said user program during said simulating and enabling generation of source code corresponding to said expressions;

means for dividing said trace file into a plurality of trace file blocks;

means for converting said trace file blocks into source code expression blocks;

means for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating including, for each source code expression block,

parameterizing said source code expression block to include at least one ~~of a loop blocking and a loop unrolling~~ optimization parameter, the at least one optimization parameter being taken from parameters of self-tuning objects corresponding to entries in a trace file block from which said source code expression block was generated,

iteratively:

selecting at least one value for said ~~respective~~ at least one optimization parameter,

compiling said parameterized source code expression block in accordance with said selected at least one value for said ~~respective~~ at least one optimization parameter, and

measuring an execution time of object code resulting from that

compiling, and,

on the basis of iteratively selecting, compiling and measuring, identifying the at least one value for said ~~respective~~ at least one optimization parameter that is associated with a ~~the~~ minimal ~~said~~ execution time for said compiled expression block; and,

means for linking said plurality of minimal timing, compiled expression blocks into said user program.